

Proposition de correction de l'examen du cours *Conception et traitement des données*

Claire Medrala (MESR), Fabien Coelho (Mines Paris – PSL)

Janvier 2025

Bravo Hobbes ! 20/20

1 Citius, Altius, Fortius – Communiter

7/7

1. Quel-le-s athlètes *français·e·s* ont gagné une médaille d'*or* dans une épreuve de *natation* ?

```
SELECT DISTINCT anom
FROM Athlete
JOIN Resultat USING (aid)
JOIN Epreuve USING (eid)
JOIN Discipline USING (did)
JOIN Medaille USING (mid)
JOIN Pays USING (pid)
WHERE pnom = 'France'
      AND mnom = 'Or'
      AND dnom = 'Swimming';
```

2. Suggestions d'index pour cette requête :

```
-- 1. en partant de la discipline
-- Discipline: dnom -> did: ok, UNIQUE
-- Epreuve: did -> eid: non UNIQUE (enom, did), enom en premier
CREATE INDEX epreuve_did ON Epreuve(did);
-- Resultat: eid -> rid -> aid: ok, UNIQUE (eid, aid)
-- Athlete: aid -> pid : ok clé primaire
-- Pays: pid -> pnom : ok clé primaire
-- Medaille: mid -> mnom : ok clé primaire

-- 2. en partant de Pays
-- Pays: pnom -> pid: ok, UNIQUE
-- Athlete: pid -> aid
CREATE INDEX Athlete_pid ON Athlete(pid);
-- Resultat: aid -> rid: non, UNIQUE (eid, aid)
CREATE INDEX Resultat_aid ON Resultat(aid);
-- Epreuve: eid -> did: ok PK
-- Discipline: did -> dnom: ok PK
-- Medaille: mid -> mnom: ok PK

-- 3. en partant de Medaille (mais pas assez sélectif)
-- Medaille: mnom -> mid: ok, UNIQUE
-- Resultat: mid -> rid -> eid/aid/mid
CREATE INDEX Resultat_mid ON Resultat(mid);
-- Athlete: aid -> pid (PK) et Pays: pid -> pnom (OK)
-- Epreuve: eid -> did (PK) et Discipline: did -> dnom (PK)
```

3. Comment afficher par ordre alphabétique les noms des disciplines qui ont au moins 12 épreuves, avec leur nombre d'épreuves ?

```
SELECT dnom, COUNT(*) AS "Nb épreuves"
FROM Discipline
JOIN Epreuve USING (did)
GROUP BY 1
HAVING COUNT(*) >= 12
ORDER BY 1;
```

4. Comment afficher pour toutes les disciplines et tous les pays, le nombre de couleurs des médailles obtenues (de 0 à 3), par ordre des disciplines et des pays ?

```
SELECT d.dnom, p.pnom, COUNT(DISTINCT mid) AS ncols
FROM Discipline AS d
CROSS JOIN Pays AS p
JOIN Epreuve USING (did)
JOIN Athlete AS a USING (pid)
LEFT JOIN Resultat USING (aid, eid)
GROUP BY 1, 2
ORDER BY 1, 2;
```

5. Comment afficher pour toutes les athlètes médaillées leur nombre de médailles par catégorie (or, argent, bronze), en ordre décroissant des couleurs puis par ordre alphabétique ?

```
SELECT anom AS "Nom",
COUNT(*) FILTER (WHERE mnom = 'Or') AS "Or",
COUNT(*) FILTER (WHERE mnom = 'Argent') AS "Argent",
COUNT(*) FILTER (WHERE mnom = 'Bronze') AS "Bronze"
FROM Resultat
JOIN Athlete USING (aid)
JOIN Medaille USING (mid)
GROUP BY 1
ORDER BY 2 DESC, 3 DESC, 4 DESC, 1 ASC;
```

Une version moins élégante avec 3 tuples par athlète médaillé :

```
-- NOTE la base ne contient que les athlètes médaillés
SELECT anom AS "Nom", mnom AS "Couleur", COUNT(rid) AS "#"
FROM Athlete
CROSS JOIN Medaille
LEFT JOIN Resultat USING (aid, mid)
GROUP BY 1, 2
ORDER BY 3 DESC, 2, 1;
```

6. Quelles sont les épreuves individuelles mixtes (nom de la discipline et de l'épreuve) ?

Version directe

```
SELECT dnom, enom
FROM Epreuve
JOIN Discipline USING (did)
JOIN Resultat USING (eid)
JOIN Athlete AS a USING (aid)
GROUP BY 1, 2
HAVING COUNT(DISTINCT a.agenre) >= 2
AND COUNT(*) <= 4 -- la lutte a deux médailles de bronze par épreuve
;
```

En suivant les définitions

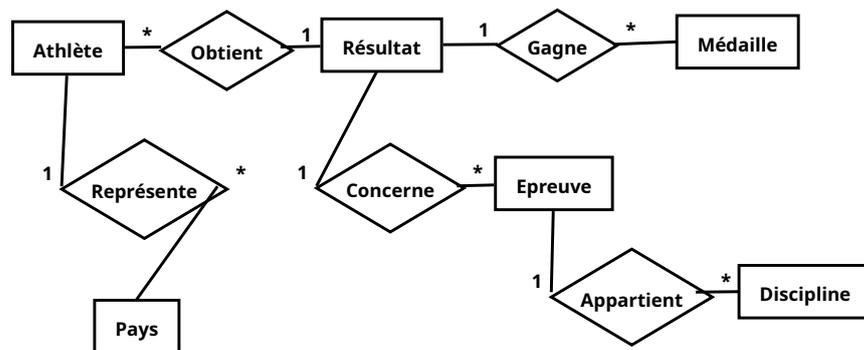
```
WITH
  EpreuveIndividuelle AS (
    SELECT e.eid, e.did, e.enom
    FROM Epreuve AS e
    JOIN Resultat AS r USING (eid)
    GROUP BY 1, 2, 3
    HAVING COUNT(*) <= 4 -- for Wrestling
  ),
  EpreuveMixte AS (
    SELECT DISTINCT e.eid, e.did, e.enom
    FROM Athlete AS a1
    JOIN Resultat AS r1 USING (aid)
    JOIN Epreuve AS e USING (eid)
    JOIN Resultat AS r2 USING (eid)
    JOIN Athlete AS a2 ON (a2.aid = r2.aid)
    WHERE a1.agenre <> a2.agenre
  ),
  IndividuelMixte AS (
    SELECT did, enom FROM EpreuveIndividuelle
    INTERSECT
    SELECT did, enom FROM EpreuveMixte
  )
SELECT dnom, enom
FROM IndividuelMixte
JOIN Discipline USING (did);
```

2 L'important est de participer !

6/6

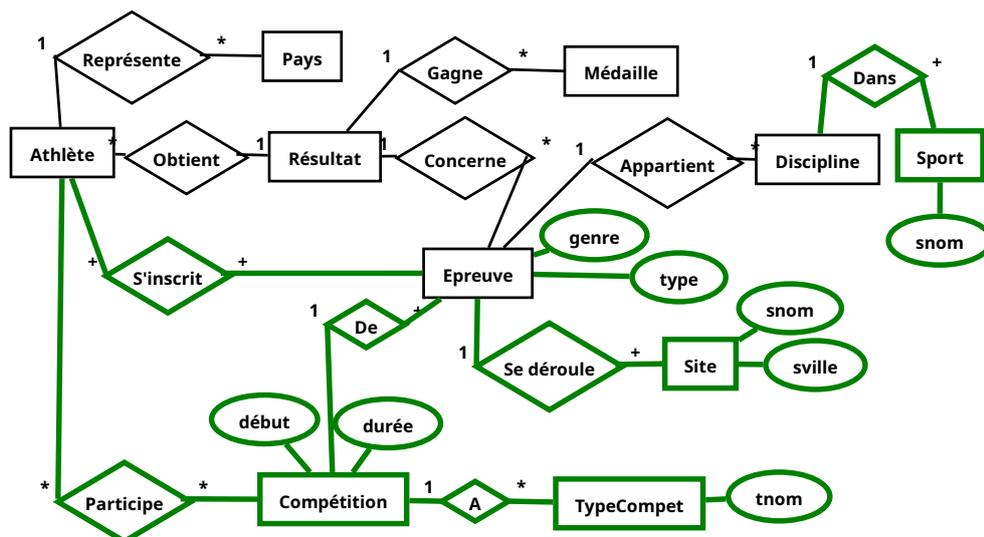
1. Reconstituer le modèle E/A du schéma relationnel précédent, **sans** les attributs existants.

Voici une proposition de modèle E/A pour



2. Ajouter les notions suivantes avec des entités, associations ou attributs supplémentaires :

Voici une proposition de modèle E/A pour



3 Vive les JO!

3/3

À partir de ce nouveau modèle E/A, donner les commandes SQL pour construire le schéma relationnel correspondant en modifiant les tables existantes et en ajoutant les nouvelles tables, en portant attention aux types des données et aux contraintes.

```

CT Sport(spId SPK, spnom TUNN)
AT Discipline(spId INNR Sport)
CT Site(sid SPK, snom TUNN, sville TNN)
CT TypeCompet(tid SPK, tnom TUNN)
CT Competition(cid SPK, debut TsNN, durée InNN, tid INNR TypeCompet, eid INNR Epreuve,
U(tid, eid))
AT Epreuve(sid INNR Site, genre CNN C( genre in ['M', 'F', 'X']), type INN C(type >= 1))
CT Inscription(aid INNR Athlete, eid INNR Epreuve, U(aid, eid))
CT Partitipation(aid INNR Athlete, cid INNR Competition, U(aid, cid))

```

4 À vos marques, prêts, partez!

4/4

Thème : ACID (Advanced Chanting In Databases)

Ce thème est très passionnant, j'ai appris plein de choses super intéressantes qui éclairent parfaitement le fonctionnement de cet aspect des bases de données à la fois théorique et pratique, et me permettent de bien comprendre les caractéristiques générales et particulières des multiples facettes de ce thème dans une perspective transversale de mise en application concrète du modèle relationnel sur des problèmes réels, tout en gardant, au delà du simple niveau fonctionnel, une maîtrise généraliste des aspects opérationnels sur Postgres qui gère le stockage à un prix compatible avec toutes les bourses, une bonne nouvelle pour nos budgets!

Il y a 3 prix Turing liés aux recherches sur les bases de données relationnelles : *Edgard Codd, James Gray et Michael Stonebraker.*